Journal of Information Processing Vol. 17 292–305 (Dec. 2009)

Regular Paper

Musicream: Integrated Music-Listening Interface for Active, Flexible, and Unexpected Encounters with Musical Pieces

Masataka Goto[†1] and Takayuki Goto[†1,*1]

This paper describes *Musicream*, a novel music-listening interface that lets a user unexpectedly come across various musical pieces similar to those liked by the user. Most existing "query-by-example" interfaces are based on similarity-based searching, so they return the same results for the same query, meaning that a user of those systems always receives the same list of musical pieces sorted by similarity. Therefore, most existing systems do not provide a user an opportunity to encounter various unfamiliar musical pieces by chance. Musicream facilitates active, flexible, and unexpected encounters with musical pieces by providing four functions: the *music-disc streaming function* which creates a flow of many musical-piece entities (discs) from a large music collection, the *similarity-based sticking function* which allows a user to easily pick out and listen to similar pieces from the flow, the *meta-playlist function* which can generate a playlist of playlists (ordered lists of pieces), and the *time-machine function* which automatically records all Musicream activities and allows a user to visit and retrieve a past state as if using a time machine. In our experiments, these functions were used seamlessly to achieve active and creative querying and browsing of music collections, confirming the effectiveness of Musicream.

1. Introduction

Enterprise services for delivering music via the Internet are increasingly widespread and continued growth in the demand for such services seems inevitable in the foreseeable future. Some advanced services provide flat-rate, unlimited music subscription services that give users unlimited on-line access to several million musical pieces [⋆1]. The most popular and basic interface for those services is based on lists of bibliographic text information such as titles, artist names, and genres provided on a display. It typically provides a function for playing back musical pieces in order and a function for searching musical pieces or artists by specifying bibliographic information. It also provides a function for browsing the whole music collection by choosing genres and a function for recommending musical pieces or artists. This text-based interface works well when a user already has in mind particular musical pieces that he or she would like to listen to. However, given that such music services offer a huge music collection, many users tend to mainly listen to familiar musical pieces for a few months and then become bored with those pieces. Thus, users will increasingly feel desires like "I want to hear something different that suits my taste" and will want to find something new by themselves when accessing flat-rate, unlimited music subscription services or large personal music collections. In addition, declining production costs have made it much easier for amateur or professional artists to produce their own music without the support of record labels and the Internet enables such artists to easily reach out to their audiences; as a result, personal music collections full of unfamiliar pieces are becoming bigger and bigger [⋆2].

The goal of our study is to provide a comprehensive integrated music-listening environment in which various browsing and querying functions can be used in an easy and seamless manner to increase opportunities to encounter unfamiliar but interesting musical pieces when actively browsing a musical collection. Although various previous studies have resulted in music-recommendation or playlist-generation schemes [1)–8)] based on collaborative (social) or content-based filtering, encounters with unfamiliar music tend to be passive on the part of listeners. While many effective similarity measures [9)–16)] have been proposed to enable users to listen to music of similar mood, most previous "query-by-example" interfaces for searching similar music only provide functions for listing musical pieces similar to a certain musical piece. Because a user will always receive the same ranked list of musical pieces given the same query and music collection, op-

†1 National Institute of Advanced Industrial Science and Technology (AIST)
∗1 Presently with The Graduate University for Advanced Studies

⋆1 For example, Napster 2.0, a popular online music subscription service, lets users listen to over eight million musical pieces for a monthly flat-rate subscription fee. It had over 830,000 paid subscribers in 2007.

⋆2 Nowadays, for example, many amateur artists submit their musical pieces to video sharing services.

portunities to encounter previously unknown musical pieces in the collection are limited. As for the music playback order, many music players already have functions enabling users to specify *playlists* (i.e., lists of musical pieces for playback). The drawback here is that the operations provided by these functions are usually limited to changing the order of pieces and to inserting and deleting pieces in existing playlists. In addition, the use of each playlist is often temporary, where a created playlist is discarded after being used several times. While it might be interesting for a user to listen to music from a past playlist (e.g., from ten years earlier), this is impossible unless the user periodically saves playlists, which is quite troublesome.

To overcome these limitations, we have developed a music-listening interface called *Musicream* ("music" + "stream") that enables a user to initiate unexpected but interesting encounters with musical pieces and provides a user with flexible interfaces for actively browsing a music collection and managing playlists. The idea behind Musicream is to see if we can break free from stereotyped ideas of how music-listening interfaces must be based on lists of musical-piece titles and artist names. Musicream applies the following functions to create a novel music-listening environment without listing musical-piece titles.

(1) *Music-disc streaming function*: lets musical-piece entities (discs) stream down one after another on a screen for user browsing and selection. This facilitates unexpected favorable encounters when the user is actively browsing part of a collection consisting of several million musical pieces: given such a huge collection, it is difficult to visualize and manage the entire collection on a screen.

(2) *Similarity-based sticking function*: attaches similar musical pieces to a selected music disc like pieces of metal stick to a magnet ⋆1. This is an active and flexible version of the useful "query-by-example" similarity search.

(3) *Meta-playlist function*: enables the user to try out various playback orders by making it easy to rearrange groups of musical pieces on the screen in a manner similar to rearranging CDs on a real-world desktop. This helps the user devise better playlists.

(4) *Time-machine function*: provides a means of moving freely back in time to restore a past Musicream screen as if using a time machine to return to a point in the past where one was listening to music. This is a version of the useful "undo" function that provides the ultimate in flexibility.

These functions make it possible, for example, to pick out a musical piece (disc) from among many streaming on the screen and then attach other pieces similar in mood to that musical piece, thereby creating a group of pieces. Furthermore, by simply continuing this process, several groups (playlists) of musical pieces can be formed and placed in a blank area on the screen in a manner that determines the playback order of those groups. In this way, users can enjoy music in a much more active manner. Moreover, all operations on Musicream are recorded automatically so that any past state can be returned to at any time. A user can reproduce what he or she was listening to at some point in the past and then continue with other operations from within that state if so desired.

This paper is organized as follows. Section 2 introduces the functions provided by Musicream and Section 3 describes how they are implemented. Section 4 presents the results of experiments on the use of Musicream. Section 5 discusses related research and Section 6 summarizes this paper's contributions.

2. Musicream Functions

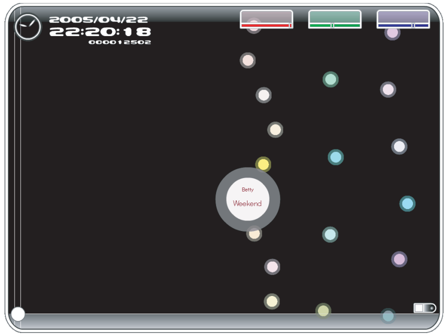
Musicream provides four novel functions (**Fig. 1**) to satisfy user desires like "I want to hear something different that suits my taste." Instead of working as independent functions, they operate in a cooperative and comprehensive manner to enable new music-listening formats. Each of these functions is described below.

2.1 Music-Disc Streaming Function

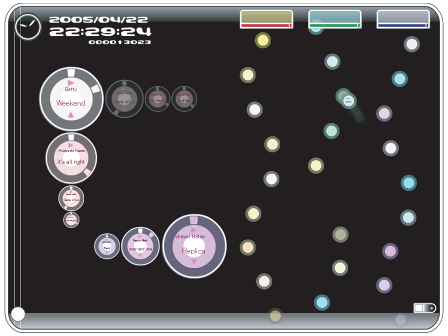
This function presents the user with images of discs, each corresponding to a musical piece in a music collection ⋆2. These discs flow from top to bottom on the screen, one after another, and the user can select a disc and listen to that musical piece. This function is especially useful as a means of encountering various musical pieces by chance as opposed to making a specific request as in "I

⋆1 The use of a magnet metaphor for an interface is also discussed in Ref. 17).

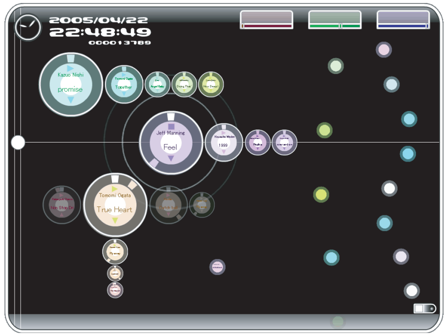
⋆2 In this paper, "music collection" means any set of musical pieces that a user is able to listen to. This would include groups of musical pieces stored on portable music players and personal computers as well as those on flat-rate, unlimited music subscription services.



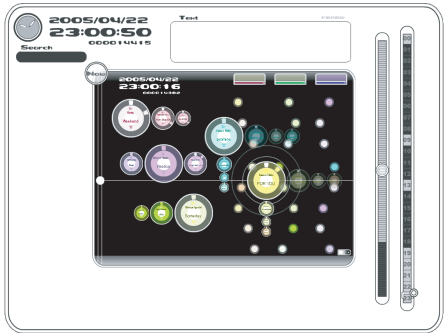
(a) *Music-disc streaming function*: Active and unexpected encounters with interesting musical pieces



(b) *Similarity-based sticking function*: Controllable and flexible "query-by-example" similarity search



(c) *Meta-playlist function*: Advanced playlist editing using a playlist of playlists



(d) *Time-machine function*: Intuitive browsing and perfect recall of past music listening activities

Fig. 1 Four functions of Musicream.

want to hear *this* musical piece."

The encountering of previously unknown music by chance is not without precedent. People often listen to music broadcast on radio and television or listen to songs or musical pieces that appear on hit charts or are recommended by friends. Various studies have been made on music-recommendation or playlist-generation schemes [1)–8)] based on collaborative (social) or content-based filtering. If a person were to listen to only musical pieces that have been selected by other people or systems, though, such encounters would tend to be passive from the user's point of view. At the same time, it is difficult to use a personal music collection consisting of thousands of musical pieces for the purpose of encountering music by chance. While music search methods based on bibliographic information such as titles and artist names have been used, they have not considered chance encounters with musical pieces [⋆1]. There are also methods based on folder-based

⋆1 Although some existing players support a random shuffle playback function (such as Apple iPod shuffle), they do not allow a user to intentionally control the general nature of music encounters, which Musicream does allow.

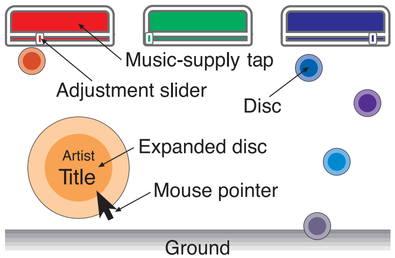


Fig. 2 Music-disc streaming function: Discs corresponding to musical pieces stream downwards from three colored taps.

hierarchical classification by music genre, artist name, etc. that allow a user to search from one hierarchical layer to another to refine the search. These methods, however, narrow down searches by promoting hierarchical selections such as "jazz ⇒ bebop," which unnecessarily decrease the possibility of unexpected but interesting encounters with musical pieces.

Our music-disc streaming function enables music encounters not likely to occur through broadcasting, recommendations, traditional searching, or hierarchical structures. **Figure 2** illustrates the Musicream screen along with some basic terminology. Each of the three rectangular boxes at the top represents a *music-supply "tap"* that releases small discs one at a time, each disc corresponding to a musical piece. These discs fall straight down at the same speed (since a tap is fairly wide, released discs are spread out crosswise along a tap). The taps are designed so that each releases musical pieces of a different mood, and the rate of release — i.e., the interval between disc releases from the same tap — can be adjusted with the sliders above each tap. A falling disc that reaches the bottom of the screen (the *"ground"*) disappears. The user may remove a disc that appears interesting from these streams of falling discs and listen to that musical piece by dragging the disc using a mouse (or a stylus, touch tablet, etc.). Rolling the mouse pointer over a small disc expands it and displays the title and name of the artist for the musical piece in question, as shown in the figure to the left.

The benefit of this interface is that a user can feel that music encounters are intentionally controlled. If one of several million musical pieces is randomly selected and presented to the user, the user would probably not feel any particular interest in it. On the other hand, with the music-disc streaming function the user deliberately chooses a disc from the streaming discs, and so is more likely to feel interest in that piece of music and listen to it. We think such active and intentional interaction can change user behavior in this way.

Each tap and disc is given a color that reflects the mood or feeling of a musical piece and each disc falls out of a tap with a similar color. In other words, similarity in color is associated with similarity in musical pieces, which means that a user who likes the musical piece of a selected disc can choose other pieces based on the color of that disc. While using the music-disc streaming function, a user will often want to perform that operation time after time, and this is where the similarity-based sticking function described next comes in.

2.2 Similarity-Based Sticking Function

With this function, the user picks up a disc previously removed from the streaming discs and drags it to touch other discs that are still streaming. The original disc can be dragged while being played back without interruption. This operation has the effect of selectively "sticking" (attaching) discs of similar mood to the original disc one after another. This function may be viewed as a "query-by-example" search for musical pieces, but it is not a search that simply presents a list of similar songs. In our similarity-based sticking function, the user collects musical pieces from streaming pieces according to the user's own choice. The important point here is that musical pieces can be encountered while the user is performing an operation similar to the way people pick up things they like in real life.

Here, the "stickiness" has been designed to depend on the similarity between two musical pieces in terms of mood (timbre similarity)[⋆1]. "Stickiness" means the ease of sticking a disc from the streaming discs. As shown in **Fig. 3**, two discs with high similarity will stick on first contact (high stickiness) whereas two discs with low similarity will not stick unless brought into contact several times (low stickiness). In this way, the range of similarity of musical pieces to be added can be easily adjusted by appropriately maneuvering the picked-up disc through

⋆1 Instead of mood, which is used in our current implementation, any state-of-the-art similarity based on content-based or collaborative filtering can be used for this function.

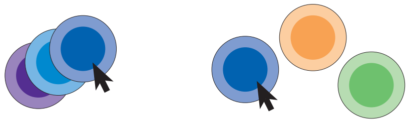


Fig. 3 Similarity-based sticking function: The "stickiness" depends on similarity in terms of mood. Similarity in color is associated with similarity in musical pieces.

Fig. 4 Two disc-series modes.

the streams of falling discs.

Sticking discs together using this function forms an overlapping series of discs as shown on the left of **Fig. 4**. This is called "*compact mode*." Clicking the top disc of this series rearranges the discs into a non-overlapping, horizontal series as shown on the right of Fig. 4. This is called "*maintenance mode*." In this mode, moving the mouse pointer over a disc expands that disc to make it the focus of that series (the discs to either side of the disc-in-focus become slightly larger). It is also possible to rearrange the disc order in the series or remove a disc from the series, as shown in **Fig. 5**.

In maintenance mode, the disc components shown in **Fig. 6** appear, thus enabling the following functions.

- *Playback control*
  Clicking on the playback button on the disc starts playback of that musical piece. If another musical piece is being played back, it is stopped right after this click (the start of playback). During playback, an animated ripple effect emanates from the disc. If the piece is played until its end, playback automatically changes to the next underlying disc. Clicking again on the button stops the playback.
- *Playback position control*

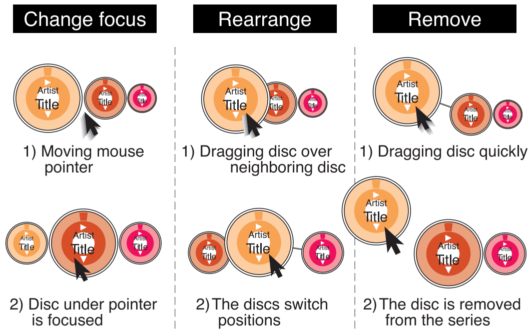


Fig. 5 Editing operations for a disc series (playlist).

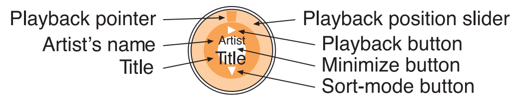


Fig. 6 Disc components.

  The playback position slider, whose function is the same as that of playback position sliders provided by ordinary music players, is placed along the periphery (circumference) of the disc. One complete loop in the clockwise direction from the twelve o'clock position corresponds to the length of that musical piece. Clicking on any position along the slider starts playback from that position in the piece.
- *Minimize*
  Clicking on the minimize button at the center of the disc switches to compact mode.
- *Sort*
  Clicking on the sort-mode button on the disc switches to sort mode as shown in **Fig. 7**. Using the disc-in-focus as a base for sorting, a copy of the other discs in the series appears below that disc in descending order of mood similarity. These duplicated discs disappear upon exiting sort mode.

In Musicream, such a disc series is already a highly functional playlist (i.e., a list of musical pieces that specifies playback order). In existing music players, "zapping" (changing from one musical piece to another in a playlist as the urge

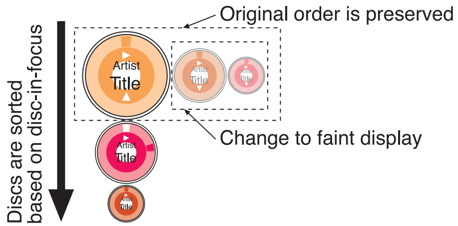


Fig. 7 Sort mode: Using the disc-in-focus as a base, other discs in the series descend in order of mood similarity.

arises) in music playback requires the user to perform a two-step procedure. First, the user must (double-)click the title of the desired selection in the playlist, and second, the user must click on the desired playback position on the playback position slider located elsewhere. In Musicream sort mode, in contrast, simply moving the mouse pointer during music playback over other discs in the series automatically starts and stops playing those musical pieces: there is no need for a new playback-start operation as long as the user stays within the same playlist. In addition, clicking on the playback position slider on the periphery of a disc right after moving the focus enables a series of operations — musical piece selection without click and playback position specification with a single click — to be performed simultaneously making for the smoothest zapping ever seen.

Similarity-based sticking enables a user to pick out discs from disc streams as desired, create one playlist after another, and leave them on the Musicream screen. Faced with multiple playlists created in this way, a user might then think about the best playback order for the playlists. This can be set using the meta-playlist function described next.

2.3 Meta-Playlist Function

Similar to rearranging a group of CDs on a real-world desktop, this function enables playlists (multiple series of discs) situated on the screen to be rearranged and their order of playback specified. This is done by treating the entire Musicream screen as a meta-playlist and playing back the playlists on the screen in order from top to bottom.

The playback pointer for the entire screen used by this function is simply a straight horizontal line named the "playback bar" as shown in **Fig. 8**. The handle at the left of the playback bar can be used to drag the bar as desired (i.e., upward or downward). Once dragged to a new position, the playback bar starts to drop (automatically moving to the bottom of the screen) and plays back the series of musical pieces in any maintenance-mode or sort-mode playlist (disc series) with which it comes into contact. A playlist in compact mode is ignored in this playback.

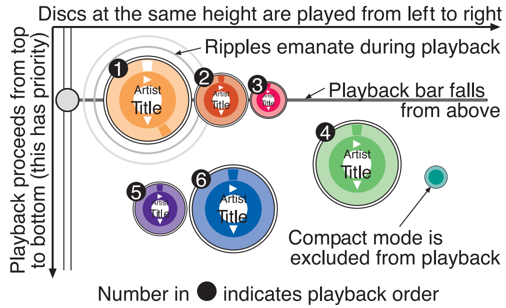


Fig. 8 Playback rule of meta-playlist function: The playback bar falls from above and plays playlists in order of contact.

For this function, only positional relationships in the screen's vertical direction affect playback order. Horizontal relationships have no effect on playback order, and the user can make good use of this characteristic to achieve flexible playback control. For example, to consider the best playback order of several playlists through trial and error, the user can arrange those playlists horizontally and then raise or lower each playlist slightly to change their vertical relationship and hear the resulting playback order as the playback bar falls. In addition, playlists that the user does not presently wish to hear can be placed in compact mode and simply left on the screen — there is no need to delete them to prevent them interfering with playback.

By allowing free insertion, removal, and rearrangement of groups of musical pieces (playlists) to create meta-playlists, the meta-playlist function enables flexible playlist editing. If this type of switching in units of groups were attempted on an existing music player, the operations of selecting and inserting multiple musical pieces would have to be repeated any number of times while remembering group boundaries. Musicream allows groups of musical pieces to be arranged

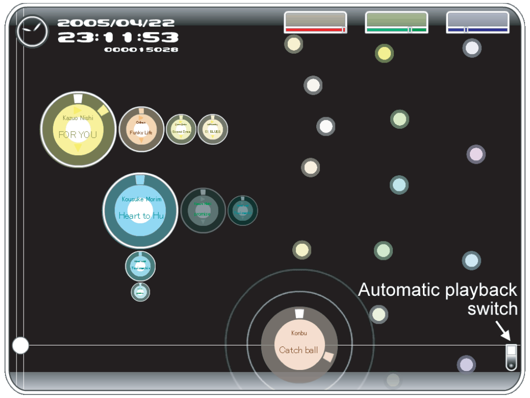


Fig. 9 Musicream screen snapshot with the automatic playback switch on: Enabling the automatic playback switch and letting the playback bar fall into the switch activates automatic playback.

on the screen as individual small playlists and allows the user to change their playback order while preserving their positional relationship in the horizontal direction. This makes intuitive trial-and-error processing possible while using positional memory. This is especially useful for making a long playlist by locally arranging the order within smaller playlists and then globally arranging the order of those smaller playlists. Playlist editing in this way is not just a method for changing playback order. It can also be viewed as a creative way of enjoying music with the user becoming actively involved[⋆1].

To facilitate the use of Musicream for playing background music, we also prepared an automatic playback mode that enables listening to musical pieces in succession without requiring user interaction. This mode is entered by turning on the automatic playback switch and letting the playback slider fall into the switch as shown in **Fig. 9**. When playback of the current musical piece ends in this mode, Musicream starts to automatically select and play back the musical piece most similar in mood to that piece from among the streaming discs on the screen (not from the playlists on the screen).

Although existing music players include a random shuffle playback function, an abrupt change in mood can easily occur from one song to the next, such as when a wild rock tune follows a romantic ballad. In the automatic playback mode of Musicream, on the other hand, the user is always listening to a new order of musical pieces while continuing to listen to pieces with a similar feel. Since the next musical piece is selected from the screen at the end of the playback of the current one, though, it is difficult for the user to predict which piece will play next.

For a user who encounters new music using the music-disc streaming and similarity-based sticking functions, and who creates playlists by trial and error using the meta-playlist function, the act of listening to music through Musicream is likely to become an everyday activity, in which case musical pieces played on Musicream will approximate that user's listening experience. Accordingly, when a user would like to know what music he or she has listened to in the past, the user should be able to find out by checking what music has been played on Musicream. The time-machine function described next makes this possible.

2.4 Time-Machine Function

This function records all operations performed by the user on Musicream as well as all screen changes so that the user can browse this record of the past and return to a desired point. This enables the user to reproduce a past Musicream screen (i.e., recall a past listening state) and continue with operations from that screen as if traveling back in time with a time machine. It is also possible to copy and paste a playlist from the past screen onto the present screen.

Reproducing a music playback order from the past in this way is difficult to achieve on existing music players. It would require that playlists be intentionally saved for future reference, but this method would only be practical if the user knew exactly what playlists would be important in the future. The time-machine function works as automatic saving or loading functions that do not require explicit commands from the user. In addition, Musicream allows the user to label the current point in time with a keyword or sentence that can then be used to later recall that point.

When the button to use the time-machine function is pushed, two time-travel

⋆1 The importance of playlists can be understood from the way that artists carefully determine song order in their albums and from the attention given to WWW sites that present personal playlists (e.g., http://www.artofthemix.org/).

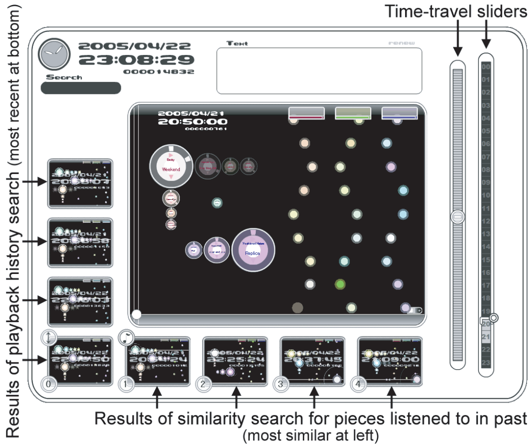


Fig. 10 Musicream screen snapshot of the time-machine function: The time-travel sliders consist of a rewind slider (left) for travel back in time in units of seconds relative to the present and a date/time slider (right) for specifying particular dates and times to return to. The results of playback history search and past similarity search are displayed when a disc is clicked.

sliders are displayed as shown in **Fig. 10**.

- *Rewind slider*
  Moving this slider takes Musicream back through time in units of seconds relative to the present. The user can browse the operation history as if he or she was rewinding recorded (video-taped) screen images.
- *Date/time slider*
  Moving this slider specifies a particular date and time (year, month, day, hour, minute) in the past. The slider is colored only for the intervals during which Musicream was operating to make it easier to find and return to a date and time during which Musicream was being used.

The time-machine function makes it even easier to listen to music in the present after deciding on a playback order through trial and error. Because past operations can be recovered at any time, the user need not hesitate to throw away a musical piece picked out through the music-disc streaming function when thinking there might be a better piece available. While it is difficult to intentionally reencounter musical pieces from the past as they stream and fall from the taps in the present, it is easy to return to a past encounter and pick them up there.

We also provide two functions to support the recalling of the past through the time-machine function. Even if specific information, such as date/time, title, etc., corresponding to a past Musicream screen cannot be remembered, these functions enable searching to be done while recalling musical pieces listened to in the past and their screens.

- *Playback history search*
  Starting with a certain musical piece, this function searches for past screens where this piece was listened to.
- *Past similarity search*
  From among all musical pieces listened to in the past, this function searches for past screens where musical pieces similar in mood to the starting musical piece were listened to.

The results of both of these functions are simultaneously displayed by clicking a musical piece in the time machine operation as shown in Fig. 10.

3. Implementation

Implementing Musicream requires preprocessing of each musical piece in a music collection and implementation of the Musicream interface which is executed with sound files of musical pieces in the MPEG Audio Layer 3 (MP3) format and a music-catalog file including the results of preprocessing. Although our current implementation supports only MP3 sound files stored on a local hard disk, Musicream can be easily applied to online musical pieces and any music subscription services.

3.1 Preprocessing

A music-catalog file in XML format is generated from MP3 sound files in a music collection. The file includes title, artist name, name of the MP3 sound file, feature vector for computing music similarity, and associated disc color for each musical piece. The feature vector is extracted from each musical piece and the disc color (hue and saturation) is assigned on the basis of the feature vector.

Although we can use any feature vector designed for computing music similarity, in our current implementation we use a 30-dimensional feature vector (described in Ref. 9)) obtained by automatically analyzing the mood of each musical piece. The effectiveness of this feature vector for genre classification has been confirmed [9]. It consists of the mean and variance of local spectral features (centroid, rolloff, flux, and zero-crossings) across the entire musical piece (eight dimensions); average values of mel-frequency cepstral coefficients (MFCC) across the entire musical piece (ten dimensions); portion of the musical piece occupied by low-energy intervals (one dimension); pitch content features reflecting periodicity in pitch (five dimensions); and rhythmic content features reflecting periodicity in beat (six dimensions). The feature vectors for all sound files are extracted using Tzanetakis's MARSYAS [18], a software framework for audio analysis and synthesis.

The disc color is determined from the color circle whose circumference and radius correspond to hue and saturation, respectively. Each musical piece is projected into the circle according to its feature vector. Principal component analysis (PCA) is used to reduce the dimensionality of feature vectors to a two-dimensional vector on a plane. The planar coordinates consisting of the first and second principal components are converted to polar coordinates, and then angle θ is assigned to hue and distance r from the origin is assigned to saturation.

3.2 Implementation of the Musicream Interface

The Musicream interface is implemented using Macromedia Flash MX Professional 2004. It takes the music-catalog file as input and provides the four functions of Musicream.

First, for the music-disc streaming function, the hue angle θ is divided into three equal 120° sections, and each section is assigned to one of the three music-supply taps. Any one tap can release only discs having hue values in the range assigned to that tap. The number of taps can be changed by assigning angle θ in different ways. The interval of disc release can be adjusted by a slider control in a range from 3 to 10 seconds, which is determined so that discs do not overlap when streaming discs are expanded.

Next, for the similarity-based sticking function, the ease-of-sticking between two discs is determined on the basis of the hue angle θ of each disc. If the difference between θ of one disc and θ of the other lies in the range from 0 to 30°, the two discs will stick on the first try. For differences of 30 to 80°, 80 to 130°, and 130° and greater, the two discs will stick on the second, third, and fourth tries, respectively.

For playlist sort mode, in which discs in a series are sorted in order of similarity and hung from the base disc, the degree of similarity between two discs is defined in terms of the cosine angle (scalar product) between the 30-dimensional feature vectors of those discs.

Finally, the time-machine function is achieved by continuously recording at one-second intervals the snapshot information needed to reproduce past states (all discs on the screen, each tap's internal state, date and time, etc.). Though it depends on the number of discs on the screen, a memory capacity of about 10 Mbytes is needed to save this information in XML format for one hour of Musicream use; there is a lot of room here, however, for compression.

4. Experimental Results

We operated Musicream using a music collection that included all 315 musical pieces of the RWC Music Database [19]⋆1 and 1572 popular songs which appeared on Japanese hit charts from 2000 to 2004. We found that the proposed interface functioned effectively and that the four Musicream functions working in combination provided an active music-listening experience that was unique. When first using Musicream, users typically tried the similarity-based sticking function after finding a musical piece that they liked in the stream of discs. After becoming more familiar with Musicream, though, instead of first determining whether they liked each musical piece they picked out, users went directly to sticking many other musical pieces to the piece they picked out and then entered sort mode where they would listen to those pieces through "zapping" and then decide which ones they liked. This was because they found that even if they did not like the first piece selected, they could still unexpectedly come across one that they did like through sticking. Furthermore, by listening to a group of musical pieces obtained through sticking, users could understand what kind of music they might like to hear at that time.

⋆1 Note that Musicream can support operations using a much greater number of musical pieces. The Musicream concept should prove highly suitable for unlimited music-listening environments, like flat-rate music subscription services, as well as personal music collections.

Table 1 Questionnaire results with regard to the four functions: 27 subjects answered eight questions using a scale from "−3" (worst) to "+3" (best).

	Number of subjects						
Subjective evaluation score	−3	−2	−1	0	+1	+2	+3
1-1. Music-disc streaming function is interesting	0	1	2	1	2	5	16
1-2. Music-disc streaming function is an effective way to encounter unfamiliar musical pieces	1	0	0	1	4	7	14
2-1. Similarity-based sticking function is interesting	0	0	0	2	2	8	15
2-2. Want to use similarity-based sticking function	0	0	3	4	6	6	8
3-1. Meta-playlist function is interesting	1	0	0	4	4	4	14
3-2. Meta-playlist function is convenient	0	2	1	3	11	3	7
4-1. Time-machine function is interesting	1	0	0	2	2	8	14
4-2. Time-machine function is convenient	0	0	0	3	5	6	13

Table 2 Comparison between Musicream and an ordinary music player with regard to the "zapping" operation: 27 subjects answered two questions using a scale from "−3" (ordinary music player) to "+3" (Musicream).

	Number of subjects						
Subjective evaluation score	−3	−2	−1	0	+1	+2	+3
1. Which makes it easier to perform a "zapping" operation?	1	3	6	3	2	5	7
2. Which is more convenient for performing a "zapping" operation?	1	0	6	3	4	7	6

4.1 User Study of Musicream

To further analyze the advantages of Musicream, we conducted a user study with 27 subjects (16 male, 11 female) who were not familiar with the music collection used here. To this end, we used musical pieces only from the RWC Music Database in this experiment. To evaluate their subjective assessment of Musicream after the subjects gained a good command of Musicream operation, each subject was asked to complete a subjective questionnaire after freely using Musicream for five minutes, excluding the time during which they received brief instructions. As shown in **Table 1**, questionnaire results with regard to the four functions indicated that more than 81.5% of the subjects rated each of the four functions as interesting (in particular, 92.6% found the similarity-based sticking function interesting), and that 92.6% of the subjects thought the music-disc streaming function was an effective way to encounter unfamiliar musical pieces such as those available through flat-rate, unlimited music subscription services. In addition, another questionnaire result indicated that 96.3% of the subjects wanted to use Musicream in the future. We also found that Musicream was easy enough to use without much training.

We then had each subject use both Musicream and an ordinary music player with conventional playback buttons and a playlist editor ⋆1 for three tasks — listening through the "zapping" operation, editing playlists, and finding music — and complete a subjective questionnaire comparing their operation. As shown in **Table 2**, the number of subjects who rated the zapping operation on Musicream as easier to use and more convenient exceeded the number who preferred these operations on the ordinary player. With regard to the meta-playlist function (**Table 3**), 96.3% of the subjects thought that the Musicream function was easier to use and more convenient than conventional playlist editing when the playback order was rearranged several times in units of groups of musical pieces. For browsing unfamiliar musical pieces to find a musical piece having a certain mood (e.g., uplifting music), the number of subjects who thought Musicream was more convenient, made music easier to find, and was more enjoyable to use than an ordinary music player was, respectively, 3.6, 3.8, and 5.75 times the number who

⋆1 The ordinary music player had buttons for start, stop, pause, fast-forward, and rewind of a musical piece, a playback slider for changing the playback position, a display of elapsed time, and a playlist editor where musical pieces can be inserted, deleted, and reordered.

Table 3 Comparison between Musicream and an ordinary music player with regard to the meta-playlist function: 27 subjects answered two questions using a scale from "−3" (ordinary music player) to "+3" (Musicream).

	Number of subjects						
Subjective evaluation score	−3	−2	−1	0	+1	+2	+3
1. Which is easier to use to rearrange the playback order?	0	0	0	1	0	1	25
2. Which is more convenient for rearranging the playback order?	0	0	0	1	0	2	24

Table 4 Comparison between Musicream and an ordinary music player when browsing unfamiliar musical pieces to find a musical piece having a certain mood: 27 subjects answered three questions using a scale from "−3" (ordinary music player) to "+3" (Musicream).

	Number of subjects						
Subjective evaluation score	−3	−2	−1	0	+1	+2	+3
1. Which is more convenient for finding a piece having a certain mood?	3	1	1	4	4	10	4
2. Which makes it easier to find a piece having a certain mood?	3	2	0	3	5	8	6
3. Which is more enjoyable to use?	2	0	2	0	1	7	15

rated the ordinary music player more highly (**Table 4**).

These results showed that Musicream was an effective way, through its four convenient functions, to enable users to browse music collections to find unfamiliar but interesting musical pieces.

4.2 Experiences with Musicream

One of the authors used Musicream on a trial basis for several weeks in 2005. For this trial, 1572 popular songs which appeared on Japanese hit charts from 2000 to 2004 were used. Musicream was executed on a tablet PC (Microsoft Windows XP Tablet PC Edition) equipped with a touch screen and a digital pen without any keyboard or mouse. All operations were therefore done using the digital pen.

All the functions of Musicream were used in general, especially at the beginning of the trial, though the music-disc streaming function and the similarity-based sticking function were used most often. When the main purpose was music listening, these functions were particularly useful since they made it easy to pay enough attention to operations and interactions.

When there were other purposes or tasks, though, and music playback was just for background music, such active interactions were sometimes too much. In such cases, or when the user got tired of the interactions, the automatic playback mode described in Section 2.3 was useful. This mode does not require any interaction and could be used seamlessly: the mode could be turned on while playing back the current piece, and when an interesting piece was found during the automatic playback, it could be taken and dragged so that other active operations could follow. By changing the slider of music-supply taps during the automatic playback, we could control the possible colors (moods) of played-back discs because this changed the streaming discs on the screen and the next piece in the automatic playback was selected from those discs.

Some drawbacks of Musicream were also found. Because the main focus of Musicream was on operations at the level of musical pieces, operations at the level of artists were not supported at all although the user sometimes anticipated their availability. In addition, there were no global search functions, which the user anticipated using to find particular musical pieces or artists.

In conclusion, the present version of Musicream is not yet perfect, though we believe it opens up new music interface possibilities. A possible future approach is to integrate Musicream with conventional text-based music playback interfaces that show lists of bibliographic information since Musicream is complementary to such conventional interfaces.

5. Related Research

To go beyond conventional interfaces based on bibliographic information or query-by-example retrieval, several interesting approaches for browsing music collections have been reported.

For example, Tzanetakis, et al.[20] developed the "GenreSpace" interface for browsing music collections in a three-dimensional space into which musical pieces are projected according to their similarity; they also developed the "GenreGram" tool for displaying, along with real-time audio input, several up-and-down cylinders corresponding to different genres. Tzanetakis, et al.[21] have also developed other interfaces, such as "Sound Sliders" which provides continuous aural feedback of retrieved pieces while a user moves sliders of music properties such as tempo and beat strength.

With an emphasis on visualization on the level of musical pieces, Pampalk, et al.[22] reported the "Islands of Music" interface that features self-organizing maps (SOMs) and projects musical pieces onto a plane. They used a metaphor of "islands" that represent self-organized clusters of similar pieces. Using a small focused collection of music by a single composer, it is possible to map musical pieces in the shape of the silhouette of its composer[23]. Another visualization technique "U-Map"[24] using a variant of SOM called Emergent SOM (ESOM) was also proposed. The Traveller's Sound Player[25] uses the Traveling Salesman algorithm to map musical pieces of a collection onto a circle and visualizes the distribution of metadata (e.g., genre and tempo) having a certain value by changing the color of the corresponding region on the circle.

As for visualization on the level of artists, Van Gulik, et al.[26] reported the "Artist Map" interface with the focus on artists and small devices. It enables users to explore and discover music collections on small devices by projecting artists into a two-dimensional space. Artists are drawn as dots in the space so that similar artists are placed close together on the basis of a modified spring-embedder algorithm. This visualization can also be used to make playlists by drawing paths and specifying regions on top of the visualization[27].

By using metadata of sound files without analyzing audio signals, Torrens, et al.[28] developed visualization techniques where musical pieces are placed in a circle, rectangle, or tree-map. When visual information related to musical pieces is available, a "collaging" technique proposed by Bainbridge, et al.[29] is also an effective way to provide leisurely, undirected interaction with a music collection. In Musicream, visual information like jacket covers can also be used effectively by displaying a jacket cover image on or near each music disc when such information is available.

Although most of the above approaches share the same goal of enabling a non-specific music search to satisfy user desires like "I want to hear something different that suits my taste", Musicream is the first interface that supports the four functions described in Section 2. In particular, the time-machine function is unique and has great potential. While the concept of time-machine computing itself was proposed for a computer desktop[30], our research is the first to discuss time-machine computing for music. In Musicream, the ability to reproduce all operations from the past produces an even greater time-travel effect. For example, when a user returns to the past and sticks a series of similar musical pieces that were not previously selected (thereby creating a new future), the musical pieces streaming on the screen revert to those that were streaming at that time. This makes it possible to unexpectedly encounter music that was popular at that point in the past.

6. Conclusion

We have described a rich, integrated music-listening interface called *Musicream* that enables seamless operation of four novel functions: "*music-disc streaming*," "*similarity-based sticking*," "*meta-playlist*," and "*time-machine*." The main contribution of Musicream is to provide a novel music-listening environment that enables a user to interact with a huge music collection in active, flexible, and creative ways, which go beyond traditional techniques of music information retrieval.

Although the basic concept of Musicream has great potential, we have not yet fully exploited it. In the future, for example, we could provide various specially designed music taps such as a tap streaming musical pieces by a specific artist (e.g., an artist name could be typed onto the tap), a tap streaming music from a hit chart at a certain time in the past or at present, or a tap streaming music selected by a famous artist or celebrity. It is also interesting to share the screen (i.e., the playlist) of Musicream with friends or anonymous users via the Internet. If they use the same flat-rate music subscription service, other users can freely listen to those pieces as well as see the titles on the screen. For the "zapping" operation, it will be useful to color chorus ("hook") sections in the playback position slider (at the disc periphery) as is done in SmartMusicKIOSK [31]: these chorus sections can be automatically detected through the RefraiD method.

Note that the concept of Musicream is also independent of similarity measures and dimensionality reduction techniques. We plan to use another similarity measure based on content-based or collaborative filtering and another dimensionality reduction technique such as the self-organizing map (SOM). Since Musicream is complementary to conventional music information retrieval techniques, their seamless integration should be the next goal. Future work will also include applying Musicream as an interface for commercial online music subscription services.

References

1) Shardanand, U. and Maes, P.: Social Information Filtering: Algorithms for Automating "Word of Mouth", *Proc. CHI '95*, pp.210–217 (1995).
2) Cohen, W.W. and Fan, W.: Web-Collaborative Filtering: Recommending Music by Crawling The Web, *Proc. WWW9* (2000).
3) Uitdenbogerd, A. and van Schyndel, R.: A Review of Factors Affecting Music Recommender Success, *Proc. ISMIR 2002*, pp.204–208 (2002).
4) Alghoniemy, M. and Tewfik, A.H.: A Network Flow Model for Playlist Generation, *Proc. ICME 2001* (2001).
5) Pauws, S. and Eggen, B.: PATS: Realization and User Evaluation of an Automatic Playlist Generator, *Proc. ISMIR 2002*, pp.222–230 (2002).
6) Logan, B.: Content-Based Playlist Generation: Exploratory Experiments, *Proc. ISMIR 2002*, pp.295–296 (2002).
7) Aucouturier, J.-J. and Pachet, F.: Scaling Up Music Playlist Generation, *Proc. ICME 2002* (2002).
8) Logan, B.: Music Recommendation from Song Sets, *Proc. ISMIR 2004*, pp.425–428 (2004).
9) Tzanetakis, G. and Cook, P.: Musical Genre Classification of Audio Signals, *IEEE Trans. Speech and Audio Proc.*, Vol.10, No.5, pp.293–302 (2002).
10) Aucouturier, J.-J. and Pachet, F.: Music Similarity Measures: What's the Use?, *Proc. ISMIR 2002*, pp.157–163 (2002).
11) Paulus, J. and Klapuri, A.: Measuring the Similarity of Rhythmic Patterns, *Proc. ISMIR 2002*, pp.150–156 (2002).
12) Foote, J., Cooper, M. and Nam, U.: Audio Retrieval by Rhythmic Similarity, *Proc. ISMIR 2002*, pp.265–266 (2002).
13) Dixon, S., Pampalk, E. and Widmer, G.: Classification of Dance Music by Periodicity Patterns, *Proc. ISMIR 2003*, pp.159–165 (2003).
14) McKinney, M.F. and Breebaart, J.: Features for Audio and Music Classification, *Proc. ISMIR 2003*, pp.151–158 (2003).
15) Tsai, W.-H., Wang, H.-M., Rodgers, D., Cheng, S.-S. and Yu, H.-M.: Blind Clustering of Popular Music Recordings Based on Singer Voice Characteristics, *Proc. ISMIR 2003*, pp.167–173 (2003).
16) Pampalk, E.: A MATLAB Toolbox to Compute Music Similarity from Audio, *Proc. ISMIR 2004*, pp.254–257 (2004).
17) Huang, H.-H., Nomura, S., Sumi, Y. and Nishida, T.: Toward Contents Management with Somatic Sensation, *Proc. 19th Annual Conference of the Japanese Society for Artificial Intelligence*, pp.1D2–02 (2005). (*in Japanese*).
18) Tzanetakis, G. and Cook, P.: MARSYAS: A Framework for Audio Analysis, *Organized Sound*, Vol.4, No.3, pp.169–175 (2000).
19) Goto, M.: Development of the RWC Music Database, *Proc. ICA 2004*, pp.I–553–556 (2004).
20) Tzanetakis, G., Essl, G. and Cook, P.: Automatic Musical Genre Classification of Audio Signals, *Proc. ISMIR 2001*, pp.205–210 (2001).
21) Tzanetakis, G., Ermolinskyi, A. and Cook, P.: Beyond the Query-by-Example Paradigm: New Query Interfaces for Music Information Retrieval, *Proc. ICMC 2002*, pp.177–183 (2002).
22) Pampalk, E., Dixon, S. and Widmer, G.: Exploring Music Collections by Browsing Different Views, *Proc. ISMIR 2003*, pp.201–208 (2003).
23) Mayer, R., Lidy, T. and Rauber, A.: The Map of Mozart, *Proc. ISMIR 2006* (2006).
24) Mörchen, F., Ultsch, A., Nöcker, M. and Stamm, C.: Databionic visualization of music collections according to perceptual distance, *Proc. ISMIR 2005*, pp.396–403 (2005).
25) Schedl, M., Pohle, T., Knees, P. and Widmer, G.: Assigning and Visualizing Music Genres by Web-based Co-Occurrence Analysis, *Proc. ISMIR 2006* (2006).
26) van Gulik, R., Vignoli, F. and van de Wetering, H.: Mapping Music in the Palm of Your Hand, Explore and Discover Your Collection, *Proc. ISMIR 2004*, pp.409–414 (2004).
27) van Gulik, R. and Vignoli, F.: Visual playlist generation on the artist map, *Proc. ISMIR 2005*, pp.520–523 (2005).
28) Torrens, M., Hertzog, P. and Arcos, J.-L.: Visualizing and Exploring Personal Music Libraries, *Proc. ISMIR 2004*, pp.421–424 (2004).
29) Bainbridge, D., Cunningham, S.J. and Downie, J.S.: Visual Collaging of Music in a Digital Library, *Proc. ISMIR 2004*, pp.397–402 (2004).

30) Rekimoto, J.: Time-Machine Computing: A Time-centric Approach for the Information Environment, *Proc. UIST '99*, pp.45–54 (1999).
31) Goto, M.: A Chorus-Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station, *IEEE Trans. on Audio, Speech and Language Processing*, Vol.14, No.5, pp.1783–1794 (2006).

(Received March 18, 2009)
(Accepted September 11, 2009)
(Released December 9, 2009)

Masataka Goto received the Doctor of Engineering degree from Waseda University, Japan, in 1998. He is currently a Leader of Media Interaction Group, Information Technology Research Institute at the National Institute of Advanced Industrial Science and Technology (AIST). He serves concurrently as a Visiting Professor in the Department of Statistical Modeling, The Institute of Statistical Mathematics, and an Associate Professor (Cooperative Graduate School Program) in the Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba. He received 24 awards, including the Commendation for Science and Technology by the Minister of MEXT "Young Scientists' Prize", DoCoMo Mobile Science Awards "Excellence Award in Fundamental Science", IPSJ Nagao Special Researcher Award, IPSJ Best Paper Award, IPSJ Yamashita SIG Research Awards, and Interaction 2003 Best Paper Award.

Takayuki Goto received the Master's degree at Graduate School of Media and Governance, Keio University, Japan, in 2007. He is currently a Ph.D. student in the Department of Informatics, School of Multidisciplinary Science, The Graduate University for Advanced Studies (Sokendai). His research interests include information retrieval and human-computer interaction. He received 2 awards, including the IPSJ Interaction 2005 Best Interactive Presentation Award.